



PROBLEM ORANGES



Pictured: a capybara in a hot spring from a japanese zoo with an orange on its head

In a zoo somewhere in Japan, the zookeepers decided to play the following game with the capybaras:

The capybara enclosure consists of N hot springs, which are numbered from 0 to $N - 1$. These hot springs are connected with $N - 1$ walkways. Each walkway connects two hot springs, and it is possible to reach any hot spring from any other hot spring by using these walkways. In other words, the capybara enclosure has the structure of a tree (that is, an undirected connected acyclic graph).

Initially, there is at most one capybara in each hot spring, however this can change during the game.

The game consists of several (possibly an infinite number of) rounds. Each round has 2 stages:

1. The zookeepers will throw an orange into one of the N hot springs. The capybaras will know in which hot spring the orange was thrown into.
2. At most one of the capybaras can move to an adjacent hot spring. After that, if there are no capybaras in the hot spring containing the orange, then the zookeepers win and the capybaras lose. Otherwise, the orange is eaten and the game continues.

An initial configuration (that is, the set of hot springs initially containing capybaras) is *safe* if the zookeepers cannot win the game in a finite number of rounds if both the zookeepers and the capybaras play optimally.

For each K from 0 to N , find the number of *safe* initial configurations that have exactly K capybaras. Since these numbers can be very large, find their remainders modulo 998244353.

■ **IMPLEMENTATION** You will have to implement the following function:

```
std::vector<int> solve(int N, std::vector<int> U, std::vector<int> V)
```

This function is called once by the grader and should return a `std::vector<int>` of length $N + 1$ containing the number of safe initial configurations (modulo 998244353) that have exactly K capybaras, for every $0 \leq K \leq N$.

The parameters for this function have the following meanings:

- N - the number of hot springs.



- U - a `std::vector<int>` of length $N - 1$ which contains the first endpoints of each of the $N - 1$ walkways.
- V - a `std::vector<int>` of length $N - 1$ which contains the second endpoints of each of the $N - 1$ walkways.

For each $0 \leq i < N - 1$, the i -th walkway connects hot springs U_i and V_i .

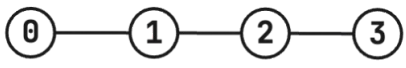
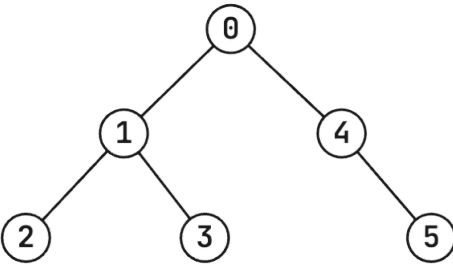
Do not forget to include the header file "oranges.h", otherwise you will get a compilation error!

- **CONSTRAINTS**
- ◆ $1 \leq N \leq 6000$
 - ◆ For each walkway (U_i, V_i) , $0 \leq U_i, V_i < N, U_i \neq V_i$
 - ◆ It is guaranteed that the given walkways form a tree (that is, an undirected connected acyclic graph).

#	Points	Constraints
1	4	There exists a hot spring which is directly connected to every other hot spring.
2	11	Each hot spring is directly connected to at most two other hot springs.
3	14	$N \leq 10$
4	9	$N \leq 20$
5	33	$N \leq 200$
6	29	No additional constraints



EXAMPLES

Input data	Output data	Explanation
1	0 1	The only safe initial configuration is {0}.
4 0 1 1 2 2 3	0 0 4 4 1	<p>The structure of the capybara enclosure in the second example:</p>  <p>There are 4 safe initial configurations with two capybaras: {0, 2}, {0, 3}, {1, 2}, {1, 3}.</p> <p>All initial configurations that have at least 3 capybaras are safe.</p>
6 0 1 1 2 1 3 0 4 4 5	0 0 0 0 11 6 1	<p>The structure of the capybara enclosure in the third example:</p>  <p>For example, the initial configuration {1, 3, 4} is not safe:</p> <p>In the first round, the zookeepers will throw an orange into hot spring 5. The capybara from hot spring 4 is forced to move to hot spring 5.</p> <p>In the second round, the zookeepers will throw an orange into hot spring 2. The capybara from hot spring 1 is forced to move to hot spring 2.</p> <p>In the third round, the zookeepers will throw an orange into hot spring 0. Since there are no capybaras that can move to hot spring 0, the zookeepers win.</p>
15 0 1 0 2 2 3 3 4 4 5 5 6 0 7 7 8 8 9 9 10 8 11 11 12 7 13 7 14	0 0 0 0 0 0 0 0 0 560 992 → 793 361 98 15 1	<p>The structure of the capybara enclosure in the fourth example:</p> 